

Web アプリケーションによる業務システムの構築

Development of Enterprise System as Web Application

石島 悌*
Dai Ishijima

(2005年6月1日 受理)

キーワード：クライアント・サーバ・システム、オープンソース、データベース、PHP、JavaScript

1. はじめに

インターネットはこの10年ほどで急速に発展し、我々の社会生活に欠かせない情報インフラに成長した。インターネットでは、電子メールのような基本的なツール以外にさまざまなサービスが提供されており、そのサービスの充実がさらなるインターネットの成長につながっている。

ネットでのサービスの代表的なものには、国税庁の確定申告などの行政サービスや、オンラインバンキングなどの金融サービス、そして、チケット予約やショッピングサイトなどのオンラインショッピングがある。これらのサービスの多くは、Web ブラウザをユーザインタフェースとした「Web アプリケーション」として実装されている。

この Web アプリケーションは、インターネット上のサービスだけでなく、インターネットの仕組みを企業内などの LAN に適用したイントラネットでも広く使われている。その代表的なものは、企業内のさまざまな情報を共有し、一元管理するための「グループウェア」である。

大阪府立産業技術総合研究所では、1996年から運用していた所内情報システムを2001年から新たに Web アプリケーションとして再構築し、2004年からその運用を開始した。本報告では、Web アプリケーションについて説明し、当研究所の業務システムの特徴を解説する。

2. Web アプリケーションとは

Web アプリケーションとは、Web (いわゆるホームページ) の仕組みを使って実現したアプリケーションソフトウェアあるいはサービスのことである。このシステムは、データを管理するサーバと、ユーザインタフェースを担当するクライアントから構成されるクライアント・サーバ型のシステムである。

従来からのクライアント・サーバ型のシステムと異なるのは、クライアント側に必要なシステムが Web ブラウザだけであり、特別なソフトウェアを必要としないことである。また、通信プロトコルに HTTP を使っているなど、インターネット・イントラネットで広く使われているリソースを十二分に活用していることもその特徴である。

これらの特徴は、システム構築に必要なコストが従来のクライアント・サーバ型のシステムよりも少なく、またシステム更新にも柔軟に対応できるというメリットにつながっている。

多くの Web アプリケーションは、図1のように、データベース、Web サーバ、Web ブラウザの3つの要素から構成されている。データベースはシステムで必要なデータを一元管理し、Web サーバは画面生成やデータ処理などを担当する。Web ブラウザはユーザへの画面表示やデータ入力などのユーザインタフェースを担当する。

インターネット上のさまざまなサービス構築において、サーバ側のシステムではオープンソースやフリーウェア

* 情報電子部 制御情報系

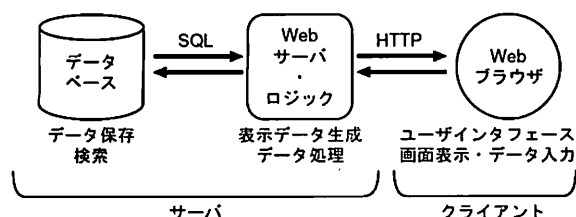


図1 3層クライアント・サーバ・システム

アが使われることが多い。当研究所の業務システムにおいても、データベースや Web サーバはオープンソースソフトウェアを活用している

3. 当研究所の業務と情報システム

当研究所では、府内企業の支援を目的として、研究、技術相談、依頼試験・機器利用（依頼試験機器および施設・設備の利用）などの業務を行っている。これらの業務をスムーズに行うために、1996年の和泉市への移転・統合以降、業務で必要となる情報をオンラインシステム¹⁾で管理してきた。

この所内情報システムは、サーバに UNIX ワークステーション、クライアントに Windows パソコンを使ったクライアント・サーバシステムとして構築された。しかし、導入後の年月の経過とともにデータの増大によってシステムの動作が遅くなってきたこと、蓄積したデータをうまく活用できないこと、システムそのものが旧式化してきたことなど、さまざまな問題が顕在化してきた。

このため、当研究所では 2001 年より新しい情報システムを検討し、柔軟性や拡張性に優れた Web アプリケーションでの構築を開始した。また、従来のシステムはそのすべてを外注としたが、新しいシステムでは、所内のスタッフがその大部分を作成した。システムの中核となるソフトウェアには、オープンソースソフトウェアを採用し、コストを大幅に削減したことも大きな特徴である。

オープンソースを用いた業務システムは、他の自治体や研究機関などでも採用事例はあるが、いずれも外注作成である。システムの仕様作成からコーディングに至るまで、その大部分を職員みずから行った例は極めて希であり、このことは当研究所のスタッフのさらなるシステム開発能力向上やノウハウの蓄積につながった。また、職員が開発を行っているので、仮にトラブルが発生しても速やかに対処できる。これはシステムの仕様変更にも柔軟に対応できるというメリットにつながっている。

4. 新しい業務システムの構成

業務システムのうち、依頼試験・機器利用システムを題材としてその構成を説明する。

(1) ハードウェア構成

業務システムのクライアントは、Windows 2000 および XP を搭載した、いわゆる「Windows パソコン」である。これらは、ユーザインタフェースを担う Web ブラウザがストレスなく動作するものであれば問題ない。現在は Pentium III 800MHz あるいは Celeron D 2.4GHz を搭載した Windows パソコンが使われている。クライアントの総数は百数十台程度である。

サーバは、2.4GHz の Xeon プロセッサ 2 基を搭載したいわゆる「PC サーバ」である。対障害性を高めるために、ストレージには SCSI RAID を採用している。

クライアントとサーバをつなぐネットワークは、1996 年の研究所移転・統合時に敷設したネットワークをそのまま利用している。なお、吹田の皮革試験所は、従来は ISDN による低速なネットワークで接続していたが、現在は光ファイバを使った IPsec VPN で接続しており、セキュリティを確保しつつ遠隔拠点においてもストレスなくシステムを使うことが可能となった。この VPN の構築・運用も研究所の職員が行っている。

(2) ソフトウェア構成

クライアントは Windows 2000 SP4 または XP SP2 上で動く Internet Explorer の 5.5 SP2 または 6.0 SP2 を使っている。ブラウザが異なるとその仕様の違い、特にブラウザで動作する JavaScript の違いから、Web アプリケーションの動作に微妙な違いが現れることがある。このため、ブラウザは Internet Explorer に限定している。しかし、実際にはできるだけ差異が目立たないように

プログラム上で工夫を行っている。このため、Mozilla Firefox などのオープンソースのブラウザでも大部分は問題なく動く。また、クライアントの OS も Windows に限定されることはなく、Mac OS や Linux などを搭載したパソコンでも業務システムを使うことが可能であると思われる。

また、一部のシステムについては、研究所内で利用している携帯電話（構内 PHS）をクライアントとして使うこともできる。この部分は、Compact HTML に準拠した携帯電話向けのページを別途用意している。

このように幅広いクライアントに対応している点も、当研究所の業務システムの特徴の一つである。

サーバでは OS に FreeBSD²⁾、データベースに PostgreSQL³⁾、Web サーバに Apache、サーバで動作す

るロジック部に PHP⁴⁾を採用してる。いずれもインターネットで広く使われ、実績のあるオープンソースソフトウェアである。

これらのオープンソースを活用しているのは、単にコストを削減するということが理由ではない。当研究所では、1996年以降、さまざまなネットワークサービスの提供⁵⁾や研究^{6,7)}をこれらのソフトウェアを活用して行ってきた。そこでの実績や信頼性が採用の理由の一つである。

(3) 異なるシステム間の同期

ここまで、依頼試験・機器利用システムの構成について説明したが、このシステムは単独で完結したものではなく、他のシステムともデータを共有している。そのデータには、職員や研究所利用者の情報がある。このマスターデータは、来所相談・研究管理システムが動いているポータルサーバに存在している。

このポータルサーバは OS に Linux、データベースに PostgreSQL, Webサーバに Resin, ロジック部に Java を採用している。依頼試験・機器利用システムのサーバとポータルサーバは、必要に応じてお互いのデータをやりとりしている。

OS やロジック部といったアプリケーションの根幹に違ったソフトウェアを採用しているが、SQL や SOAP (Simple Object Access Protocol) の問い合わせ

をうまく活用することによって、システムの利用者はサーバの違いなどを意識することはない。

このように異なるシステム間でデータを相互運用していることも当研究所の業務システムの大きな特徴の一つである。また、このようにサーバ間でデータをやりとりしている手法は、将来のシステム交換時のデータ移行にも役立つと思われる。

(4) 他のシステムでのデータ活用

本システムはデータベースに PostgreSQL を利用しているが、このデータベースでは ODBC (Open DataBase Connectivity) ドライバを使うことによって、MS Access をフロントエンドとして用いることができる⁸⁾。このため、データベースに蓄積した情報を、このシステムそのものだけでなく、Access などを使って集計するといった処理を行うことができる。

このように、蓄積された情報を他のシステムでも簡単に活用することができることも、本システムの特徴の一つである。

5. Web アプリケーションの実際

(1) 処理の概要

依頼試験・機器利用業務においては、まず、

- どのお客様が
- どんな試験あるいは機器を
- いつ、どれくらい利用するか
- その時の担当職員はだれか

という情報を確定する必要がある。次に、この情報に基づき、

- 受付票発行
- 会計処理
- 領収書発行

を行う。そしてその後、依頼試験を実施したり機器を利用していただくことになっている。さらに必要に応じて、試験報告書などを作成するなどして、一連の業務が完結する。

これらの一連の処理をプログラムとして作成することになるが、このうち、情報の確定処理をブロック図で表現すると、図2のようになる。

この処理は大きく入力部分と登録部分の二つに分けることができる。そのため、この処理は二つのプログラムとして実装することとなる。

入力部分は、機器利用を例にすれば図3のようになる。この画面では、利用者や機器、使用日、担当職員など多くの項目を入力する。しかし、実際にはそのすべてを一から入力する必要はない。項目によっては画

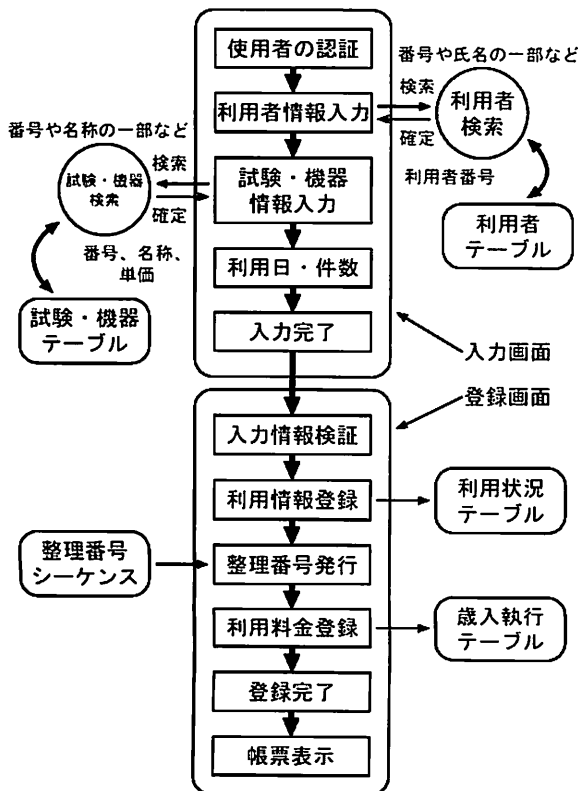


図2 情報確定処理

面が表示される時点であらかじめ値が設定されている。職員は、空欄となっている項目や修正が必要な項目だけを入力すればよい。

機器の利用料金は、使用日および時間を入力すると即時に計算される。

すべての項目を入力し、登録ボタンを押すと図4の登録完了画面が表示される。もし、未入力の項目があったり、入力項目間で矛盾があった場合は、それを修正するようにメッセージが表示される。

(2) プログラムの概要

この図3、図4の二つの画面は、それぞれ一つずつのプログラムとして実装している。次にこれらのプログラムについて説明する。

(A) 入力画面のプログラム

入力画面では、あらかじめサーバ側で次の処理を行う。

- システム使用者の認証を行う

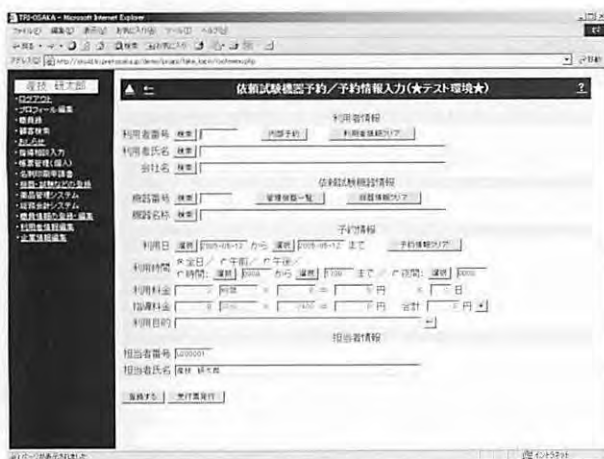


図3 機器利用入力画面

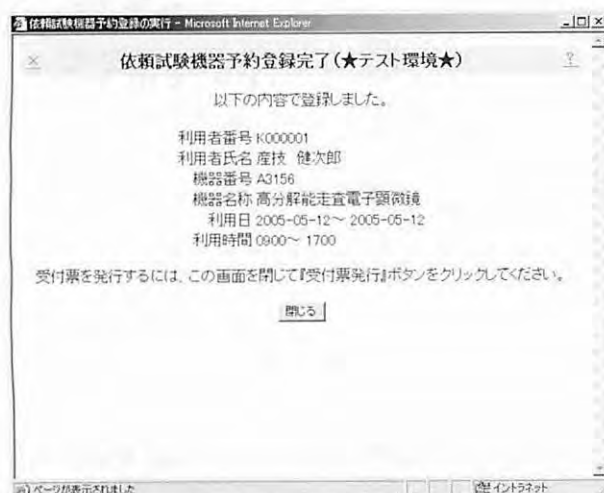


図4 機器利用登録完了画面

- あらかじめ入力する必要のない項目について、データベースを検索するなどして値を設定する
- これらの処理は PHP で記述する。なお、認証については前述のとおり、ポータルサーバへ SOAP で問い合わせを行うようにしている。

次に、入力のためのフォームと設定済みのデータをブラウザに渡す。これらは通常の Web ページ同様に HTML で記述されている。設定済みのデータなど、ページが表示されるごとにその内容が異なる部分は PHP で HTML を生成している。ブラウザで処理するための JavaScript も HTML に埋めこまれて渡される。

ブラウザは通常のページと同様に、サーバからのデータを受信し、それを表示する。ここからの処理はブラウザが担当する。

ブラウザは画面の表示後、ユーザからの入力に応じて各項目に値を設定する。また、画面上のボタンが押された場合は、JavaScript であらかじめ定義された処理を行う。料金の計算も JavaScript で行っている。

登録ボタンが押されると、JavaScript を使って未入力項目などをチェックし、問題がなければ、登録画面を生成する PHP プログラムに各項目の値を引き渡す。

以上の処理は、一つのファイルに図5のように書き込まれている。

このファイルでは、先頭部にサーバ側で行う処理が PHP で記述されている。その次に、クライアントで行う処理が JavaScript で定義されている。ファイルの中央部では、画面に表示される内容が HTML のフォームで定義されている。ファイルの末尾には、画面が表示された直後にクライアントで行う処理が記述されている。

入力画面はファイルの中では、図6のような形式で



図5 ファイルの構成

```

<TR WIDTH="100%">
  <!-- 図7の1行め -->
  <TD COLSPAN="2" ALIGN="CENTER" BGCOLOR="#80fff0">
    依頼試験機器情報
    <INPUT TYPE="hidden" NAME="kiki_search_mode"
      VALUE="<?php print($kiki_search_mode); ?>"
    >
  </TD>
</TR>
<TR>
  <!-- 図7の2行め -->
  <TD ALIGN="RIGHT" NOWRAP BGCOLOR="#80fff0">
    機器番号
  </TD>
  <TD ALIGN="LEFT" WIDTH="100%" BGCOLOR="#ffffff">
    <INPUT TYPE="button" VALUE="検索"
      onClick="search_kiki_by_no(this.form);">
    <INPUT TYPE="TEXT" NAME="kiki_no"
      SIZE="8" MAXLENGTH="6"
      VALUE="<?php printf("%s", $kiki_no); ?>"
      STYLE="ime-mode: 'inactive';"
      onFocus="next_focus('kiki_nm');"
      onChange="next_focus('button_start_date');
        search_kiki_by_no(this.form);">
    <INPUT TYPE="BUTTON" VALUE="管理機器一覧"
      onClick="search_kiki_by_kanri(this.form);">
    <INPUT TYPE="BUTTON" VALUE="機器情報クリア"
      onClick="clear_kiki(this.form);">
  </TD>
</TR>
<TR>
  <!-- 図7の3行め -->
  <TD ALIGN="RIGHT" NOWRAP BGCOLOR="#80fff0">
    機器名称
  </TD>
  <TD ALIGN="LEFT" WIDTH="100%" BGCOLOR="#ffffff">
    <INPUT TYPE="button" VALUE="検索"
      onClick="search_kiki_by_nm(this.form);">
    <INPUT TYPE="TEXT" NAME="kiki_nm"
      SIZE="80" MAXLENGTH="80"
      VALUE="<?php printf("%s", $kiki_nm); ?>"
      STYLE="ime-mode: 'active';"
      onFocus="next_focus('button_start_date');"
      onChange="next_focus('button_start_date');
        search_kiki_by_nm(this.form);">
    <INPUT TYPE="hidden" NAME="shiyotani_kb"
      VALUE="<?php print($shiyotani_kb); ?>"
  </TD>
</TR>

```

図6 入力画面の記述

依頼試験機器情報			
機器番号	検索	管理機器一覧	機器情報クリア
機器名称	検索		

図7 入力画面表示例

記述している。これは、機器の情報を入力する部分(図7)である。

ここでは、画面のレイアウトにはHTMLのTABLEタグを使っている。入力項目は、INPUTタグを使っている。また、あらかじめ値の設定されている項目には、PHPでVALUE属性に値を埋めこんでいる。

```

// 検索画面を呼び出す
function search_kiki(f) {
  var search_page = "search_kiki.php";
  // 引き渡すパラメータの設定
  var search_url = search_page
    + '?onclose=close_search_kiki'
    + '&mode=' + f.kiki_search_mode.value
    + '&kiki_no=' + f.kiki_no.value
    + '&kiki_nm=' + escape(f.kiki_nm.value)
    + '&shokuin_no=' + f.shokuin_no.value;
  // ウィンドウをポップアップ
  var newwin = window.open(search_url,
    "search_kiki_no", winstyle());
  newwin.focus();
}
// 番号で機器を検索する。「機器番号」の[検索]ボタン
function search_kiki_by_no(f) {
  f.kiki_search_mode.value = 'kiki_no';
  search_kiki(f);
}
// 名称で機器を検索する。「機器名称」の[検索]ボタン
function search_kiki_by_nm(f) {
  f.kiki_search_mode.value = 'kiki_nm';
  search_kiki(f);
}
// [管理機器一覧]ボタン
function search_kiki_by_kanri(f) {
  f.kiki_search_mode.value = 'kanri';
  search_kiki(f);
}
// ポップアップウィンドウが閉じられた時の処理
function close_search_kiki() {
  // 利用料金を計算する
  calc_kingaku(document.forms[0]);
}
// [機器情報クリア]ボタン
function clear_kiki(f) {
  f.kiki_no.value = '';
  f.kiki_nm.value = '';
  f.shiyotanka_pr.value = 0;
  calc_kingaku(f);
}

```

図8 JavaScript関数

「検索」などのボタンを押した場合は、onClick イベントを使って、あらかじめ定義しておいたJavaScript関数を呼ぶようにしている。たとえば、機器検索の場合は、図8の関数を呼ぶようになっている。この関数は、JavaScriptの機能を使ってポップアップウィンドウを表示し、その中に検索画面を呼び出している。検索画面を呼び出す際には、入力画面から必要なパラメータを送信し、検索画面においても使用者が一から情報を入力しなくてもよいようにしている。

(B) 登録画面のプログラム

登録画面では、入力画面で設定された項目をパラメータとして受けとり、各項目をチェックして問題なければ、データベースのテーブルに登録するという処理を行っている。

プログラムの構成は図5と同様であるが、処理の大部分はサーバ側での処理なので、ファイルに記述している内容の大部分は、ファイル先頭部のサーバで処理を行うPHPの記述となる。

登録処理で重要なのは、

- 入力されたデータの検証
- データベースへの登録

である。

このため、PHP 記述のうち前半分ほどはデータの検証処理になる。もし、ここでデータの不整合などが見つかれば、それをエラーメッセージとして表示する HTML を生成する。

PHP 記述の後半は、データベースへの登録作業である。ここでは、一連の処理を COMMIT~END のトランザクションブロックとして定義している。これは、データベース登録で何らかの問題が発生してもデータに矛盾を生じさせないために必要である。

依頼試験・機器利用のシステムは、日々直接お客様に関連する処理を行っているため、不具合などが発生することは許されない。このため、この登録処理は細心の注意を払ってプログラミングを行う必要があった。

(3) ユーザインタフェース

業務システムにおいて大切なことは、そのシステムを導入することによって、いかに日々の業務がスムーズに流れるようになるかという点である。つまり、システムが利用者にとって使い勝手がよいか重要である。

このシステムの画面デザインにおいては、データ入力の際には画面の上から下へと順に入力するように設計し、入力項目と関連するボタンなどはその項目の隣に配置するなど、使用者が無駄な動作をしなくても済むような配慮を行った。これにより、直感的な操作が可能となり、操作ミスを減らしたり、使用者が操作にとまどうことができるだけ少なくなるようにした。

また、使用者が入力しやすいように、いわゆる全角文字で入力する項目では IME が有効になり、半角文字の場合は無効になるように、ime-mode スタイルを使っている。また、onFocus や onChange イベントを利用して、ある項目を入力したら、自動的に次に入力すべき項目にフォーカスが当たる(カーソルが移動する)といった細かな配慮も行っている。

さらに、所内で運用している掲示板システムなどを通して寄せられた改良案や要望についても、できるだけ対処するようにした。これは職員がプログラミングや運用に携わっているから可能となったことである。

(4) 他のシステムの構成

ここまで機器利用の予約システムを例にプログラムの内部を解説したが、ほぼ同様の構成で他のシステムもプログラムを記述していった。作成したシステムは、

- 機器利用予約システム

- 依頼試験登録システム
- 施設予約システム
- 共同利用機器予約システム
- 技術研修生登録システム
- 情報検索登録システム

の合計 6 システムである。

6. システムの評価

この業務システムは 2004 年 6 月から運用を開始し、運用開始後約 1 年が経過した。これまでに大規模なシステム障害などもなく、順調に稼働している。一方で小さなトラブルはいくつか発生した。それらは、プログラムのミスや、プログラム作成者が想定していなかった操作による。しかし、この業務システムは、所内で開発を行っているため、それらの不具合についても速やかに対処することができ、不具合が大きなトラブルとなることはなかった。

システムが順調に稼働しているのは、いくつかの要因があげられる。それは以下のとおりである。

- システムの規模が大きすぎない
- 過去のシステムの遺産をうまく受け継いだ
- 開発者もシステムの使用者である

システムの使用者数は研究職員である百数十名程度で、処理する件数も依頼試験が年間 6,000 件程度、機器利用が 8,000 件程度と規模はさほど大きくはない。もし、もっと大規模なシステムであったら、大量のデータを扱うための工夫を凝らしたり、商用のデータベースシステムを導入する必要があったかもしれない。

また、データベースの構造や画面の基本デザインなども過去のシステムから流用することができ、これは開発期間の短縮につながった。

当研究所では、1996 年より業務のオンライン化を行っていた。このため、職員はオンライン処理になれており、今回のシステム導入に関して混乱が発生することはなかった。

さらにこのシステムでは、所内で運用している掲示板などをとおして得られた使用者からの意見を基にシステムの修正を行っている。使用者からの要望に応じて改良した結果、過去のシステムや他のシステムに比べて非常に使いやすくなったという評価を得ている。

7. おわりに

本報告では、まず Web アプリケーションについて説明し、2004 年 6 月より運用を開始した本研究所の業務

システムの特徴を解説した。この業務システムは拡張性や柔軟性に富んだ Web アプリケーションとして実装されており、そのシステムはインターネットで広く用いられているオープンソースを活用して作られている。また、OS やロジック記述などが異なるシステム間でうまく連携をとっている点もその特徴の一つである。

オープンソースを用いた業務システムは、最近ではさほど珍しいものではないが、本研究所のシステムは、その仕様策定からプログラムの作成、そして運用に至るまでその大部分を研究所の職員が行っているという点が大きな特徴である。使用者でもある職員みずからが作成や運用に携わっていることは、トラブルへの対処、システム変更の容易さ、ユーザインタフェースの向上などに非常に役立っている。今後、このシステムの作成および運用などで得られた知見を、さまざまなチャンネルを通して、府内企業へ技術移転を図る予定である。

最後に、本業務システムは、インターネットで公開されているさまざまなオープンソースおよびフリーウ

ェアの存在なしには完成しなかったものであり、それらの開発者とコミュニティに謝意を表します。

参考文献

- 1) 杉原俊介：大阪府立産業技術総合研究所平成 8 年度研究発表会要旨集, (1996) p.90
- 2) Michael Lucas 著, 佐藤広生 監：Abusolute BSD ~ FreeBSD システム管理とチューニング, 毎日コミュニケーションズ (2004)
- 3) 石井達夫：改訂第 3 版 PC UNIX ユーザのための PostgreSQL 完全攻略ガイド, 技術評論社 (2001)
- 4) 堀田倫英, 石井達夫, 廣川 類：PHP4 徹底攻略 改訂版, ソフトバンク (2002)
- 5) 中西 隆, 石島 梯：大阪府立産業技術総合研究所平成 8 年度研究発表会要旨集, (1996) p.24
- 6) 堀 一成, 青野繁治, 藤家洋昭, 石島 梯, 脇田由実, 高階美行：情報処理学会第 65 回全国大会, 講演論文集第 5 分冊(2003) p.347
- 7) 堀 一成, 前田 彩, 石島 梯：第 1 回情報科学技術フォーラム (FIT2002), 講演論文集 第 2 分冊 (2002) p.95
- 8) 谷口豊盛：Windows ユーザのための PostgreSQL 導入活用ガイド, 技術評論社 (2002)